

Apellidos:

Nombre:

Matrícula:

---

# Examen

## 105000016 - Programación para Sistemas Grado en Ingeniería Informática. Grado en Matemáticas e Informática

Lenguajes y Sistemas Informáticos e Ingeniería de Software

Facultad de Informática

Universidad Politécnica de Madrid

Curso 2013/2014 - Julio 2014

### Normas

- El examen puntúa sobre **12 puntos**.
- La duración total del mismo es de **una hora**.
- Se deberá tener el DNI o el carnet de la UPM en lugar visible.
- No olvidar rellenar **apellidos, nombre y número de matrícula** en cada hoja.
- La solución al examen se proporcionará antes de la revisión.
- La fecha prevista de publicación de calificaciones es el **8 de julio**, y se realizará a través del Aula Virtual de la asignatura.
- La revisión del examen tendrá lugar el **10 de julio** a las 12:00

### Cuestionario

- (1 punto) 1. ¿Qué orden muestra tu directorio de trabajo actual?  
A. passwd   **B. pwd**   C. pd   D. ls
- (1 punto) 2. Una de las siguientes órdenes no cambia el directorio actual a tu directorio personal (HOME). Señalar la respuesta correcta:  
A. cd \$HOME   **B. cd ~**   C. cd   D. cd ~
- (1 punto) 3. En el manual de Bash se puede leer  
Una lista de órdenes AND tiene la siguiente forma  
orden1 && orden2  
la orden2 se ejecuta si y sólo si la orden 1 devuelve un valor de salida (exit status) de 0  
Supóngase que un directorio sólo hay un fichero, que se llama hola. Cuál es la salida estándar del siguiente mandato:



ls adios && echo hola

A. Nada B. adios C. hola D. adios hola

- (1 punto) 4. ¿Cuál es el significado de las siguientes variables especiales en una función o en un *script* Bash?

■ \$0

**Solución:** Nombre de la orden con la que se invoca el *script*.

■ \$#

**Solución:** Número de argumetos con los que se invoca el *script*.

■ \$@

**Solución:** Todos los argumentos del *script* separados por espacios.

■ \$?

**Solución:** Valor de salida (exit status) del último mandato ejecutado.

- (1 punto) 5. Escribe la salida que produciría la ejecución del siguiente programa

```
#include <stdio.h>
void main() {
    int i, data[10];
    for (i=0;i<10;i++)
        data[i]=10-i;
    printf("%d_ %d_ %d", data[2],data[0],data[3]+data[9]);
}
```

**Solución:** 8 10 8

- (1 punto) 6. Define una estructura en C que permita almacenar una cadena de hasta 20 caracteres y un número entero. Declara una variable del tipo estructura que acabas de definir.

**Solución:**

```
struct ejemplo {
    char cadena[21];
    int numero;
};

struct ejemplo variable;
```



Apellidos:

Nombre:

Matrícula:

---

- (1 punto) 7. Escribe los comandos gdb necesarios para detener la ejecución de un programa en la línea 60 y cuando cambie el valor de la variable var1

**Solución:** break 60  
watch var1

- (1 punto) 8. Escribir como compilarías un programa en C cuyo ejecutable debe llamarse ejemplo.exe y que está compuesto por dos ficheros fuentes: ejemplo.c y ayuda.c. La compilación debe incluir información para una posterior depuración del programa.

**Solución:** gcc -g -o ejemplo.exe ejemplo.c ayuda.c

- (1 punto) 9. Sea la siguiente función:

```
void doble( int *p ) {  
    *p = *p * 2;  
}
```

Suponga que una variable 'n' ha sido declarada de la siguiente forma:

```
int n;
```

Indique la llamada a doble con el objetivo de duplicar el contenido de la variable n.

**Solución:**  
doble( &n );

- (1 punto) 10. Indique la salida del siguiente programa:

```
#include <stdio.h>  
  
int main( int argc, char ** argv ) {  
  
    int i = 1;  
  
    while( i < 5 ) {  
        if ( ( i % 3 ) == 0 ) break;  
        printf( "%d\n", i++ );  
    }  
    return( 0 );  
}
```



**Solución:**1  
2

(1 punto) 11. ¿Son equivalentes las siguientes declaraciones?

```
char cadena[] = "Hola";  
char cadena[] = { 'H', 'o', 'l', 'a' };
```

**Solución:** No.

(1 punto) 12. Sea el siguiente fragmento de código:

```
#include <stdlib.h>  
#include <stdio.h>  
  
int main( int argc, char **argv ) {  
    int **p;  
  
    if ( ( p = (int **) malloc( sizeof( int ** ) ) ) ) {  
        if ( ( *p = ( int * ) malloc( sizeof( 5 * sizeof ( int ) ) ) ) ) {  
            printf( "...\\n" );  
        }  
        else { exit( 1 ); }  
    }  
    else { exit( 1 ); }  
    return( 0 );  
}
```

Indique el tamaño en bytes de la memoria dinámica asignada en dicho fragmento. Suponga que las peticiones de memoria dinámica han tenido éxito, y que los tamaños en bytes de un puntero y de un entero son, respectivamente, 8 y 4.

**Solución:** 28